



International Journal of Marketing Management

ISSN 2454 - 5007



www.ijmm.net

Email ID: editor@ijmm.net , ijmm.editor9@gmail.com

Score-to-audio music performance synthesis is the focus of Deep Performer.

¹ Mr.Naushad Alam, ²Shaik Ramzan, ³ Mohammed Abdullah, ⁴ Shaik Noman Ahmed.

¹Assistant Professor, Department of CSE-AIML, Lords Institute of Engineering & Technology.

^{2,3,4} Student Department of CSE-AIML, Lords Institute of Engineering & Technology.

ABSTRACT

Synthesizing a musical score into an authentic performance is the goal of music performance synthesis. This article introduces the Deep Performer, a new method for converting musical scores into audio performance synthesis, by drawing on recent developments in text-to-speech synthesis. Unlike speech, music typically incorporates polyphony and lengthy notes. Therefore, we provide two novel approaches to handle polyphonic inputs and provide granular conditioning in a transformer encoder-decoder paradigm. We provide a novel violin dataset that includes paired recordings, scores, and predicted alignments to train our proposed system. We demonstrate that the model we have suggested is capable of synthesising polyphonic and harmonically structured music. In a hearing test, we outperform the baseline model—a conditional generative audio model—in terms of noise level, pitch accuracy, and timbre. On top of that, when tested on an existing piano dataset, our suggested model achieves far better overall quality than the baseline.

Index Terms— Audio synthesis, computer music, music information retrieval, machine learning, neural network

1. INTRODUCTION

In music synthesis, the mechanics of an instrument and the skill of playing it come together in a complicated way. It remains tough for a computer to synthesize a natural performance for various reasons. First, it requires a computational model for interpreting and phrasing a musical score. Second, it requires either an explicit or implicit model of the physics and acoustics by which a musical instrument sounds. Third, you need to be familiar with various approaches to playing a musical instrument. Our goal in this study is to use a data-driven strategy using machine learning to tackle all three of these difficulties simultaneously, whereas most current solutions only handle one of them. As shown in Figure 1, we introduce the Deep Performer, a unique three-step method for converting musical scores into audio files. Several methods for studying music synthesis have been previously investigated. While previous work has concentrated on creating accurate musical note samples [1-3], our goal here is to recreate the whole performance. As a starting point, we will use one method for music synthesis that involves training generative audio models using aligned piano rolls [4, 5].

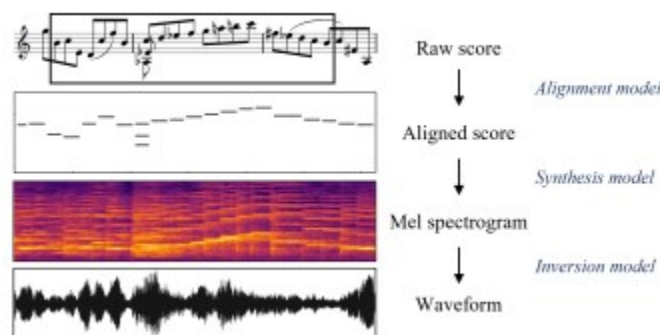


Fig. 1: The three-step process for converting musical scores into audio for live performance is outlined here.

retrieved from a recording, either from the lyrics or the demo singing audio, or from both, are the fundamental frequency (F0) contour and the loudness curve [6,7]. However, there are many that use neural networks to derive

dynamic and expressive timing from raw scores [9]. Our suggested system's inversion model, Hifi-GAN [13], is one of many that have investigated the problem of reversing mel spectrograms to waveforms [10–13]. We were unable to find any previous work on music synthesis using deep neural networks that did not either accept monophonic (i.e., one pitch at a time) inputs or required expressive timed inputs [4-8, 14-16]. Unaligned polyphonic scores have never been accepted as inputs before our suggested approach. Given the similarities between text-to-speech (TTS) and score-to-audio synthesis, we take inspiration from recent developments in TTS synthesis [18] and apply them to music synthesis. Our three-stage approach for score-to-audio music synthesis is designed to address these similarities. Music synthesis varies from speech synthesis, despite their similarities, due to the prevalence of polyphony and extended notes in music. Our proposed new polyphonic mixer aligns the encoder and decoder in a transformer encoder-decoder network [19,20], allowing it to handle polyphonic music. We suggest a novel note-wise positional encoding to train the model to respond differently at note beginnings, middles, and ends, thereby providing fine-grained conditioning. Our new dataset contains 6.5 hours of high-quality violin recordings, scores, and predicted alignments; it fills a gap in the existing literature that addresses the challenge of training score-to-audio music synthesis models. Our trials provide quantitative and qualitative evidence of the efficacy of our suggested approach. Finally, we perform a subjective hearing test to evaluate our proposed model to a baseline model that employs Hifi-GAN [13] to synthesis the waveform directly from an aligned piano roll. On our project website, you may get audio samples.

2. METHODS

The three-stage system that we propose for score-to-audio music synthesis is shown in Fig. 1. It comprises an alignment model that predicts the expressive timing of each note in a musical score, a synthesis model that creates the mel spectrogram from the aligned score, and an inversion model that uses the synthesized mel spectrogram to generate the audio waveform.

2.1. Alignment model

A transformer encoder receives a note sequence and tempo as inputs into the alignment model, and a fully-connected layer produces the beginning and ending times of each note as outputs. The output alignment is in frames, but the input score is in metric time with a musically meaningful unit, like quarter notes. The pitch, duration, and (optional) velocity of each note are defined. To further aid the model in learning the various artists' playing styles, we also include their IDs. By training the alignment model to reduce the mean squared error (MSE) between the ground truth and predicted onsets and durations, in frames, we may improve its accuracy.

2.2. Synthesis model

Given the resemblance between TTS and score-to-audio synthesis, we suggest a transformer encoder-decoder model for our synthesis model based on [21]. According to [21], the encoder's text embeddings are first length-awaredly extended, and then their lengths are combined to create the frame embeddings that the decoder uses. This is termed the state expansion mechanism [21, 22]. But polyphony is common in music, unlike in speech. We suggest the polyphonic mixer as a means to deal with inputs that are polyphonic. Figure 2 shows how the encoder begins by embedding a series of notes from the input notes. After that, the polyphonic mixer takes each note embedding for a given frame and adds them up based on when they start and how long they last. This creates a series of embeddings for the frame. At last, the decoder reads the embedded frames from the mel spectrogram sequence. The output vectors in the state expansion method [21, 22] stay the same throughout a note, and positional information is absent from each note. We contend, however, that the model's ability to distinguish between the start, middle, and finish of a note depends on such note-wise positioning information. As a result, our note-wise positional

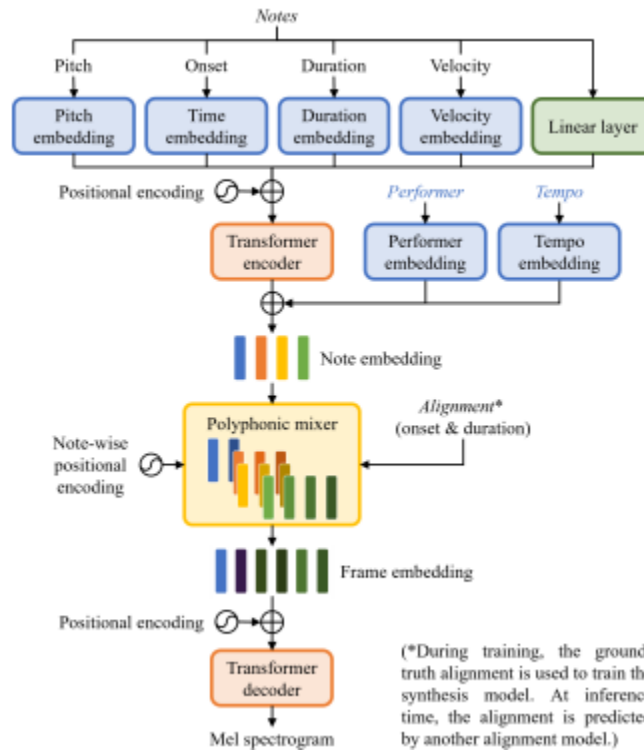


Fig. 2: The suggested synthesis model, depicted graphically.

encoding to offer a fine-grained conditioning to the decoder. In mathematics, the relative location inside a note is denoted by $k \in [0, 1]$. We have the corresponding frame embedding at position k for a note embedding v_{note} as $v_{frame} = (1 + kw) v_{note}$, where w is a learnable vector initialized to small random numbers so that v_{frame} is initially less than or equal to v_{note} . After training, the synthesis model's goal is to reduce the logarithmic mean square error (MSE) between the synthesized mel spectrograms and the source data.

2.3. Inversion model

Waveform synthesis from mel spectrograms has been the subject of previous research [10–13]. As our inversion model, we use the cutting-edge Hifi-GAN model [13] in our study. We take notice of the fact that we can train the models using various datasets thanks to the suggested three-stage workflow. For instance, if unaligned data are more readily available, a bigger dataset might be used to train the inversion model instead of aligned data.

3. DATA

For the purpose of training our proposed system, we construct a new dataset consisting of 6.5 hours of professional violin recordings, scores, and estimated alignments as there is currently no such dataset that offers matched audios and scores with fine alignments. Due to copyright concerns, we have settled on Bach's solo violin sonatas and partitas (BWV 1001–1006) because of how easily accessible they are.

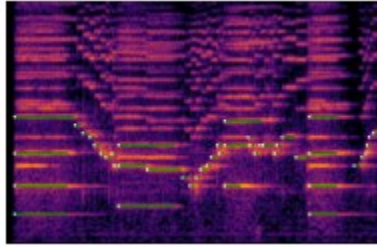


Fig. 3: Using the first twenty seconds of a violin recording as an example, here is the constant-Q spectrogram, with the white dots representing the predicted onsets and green lines representing the durations. web-based public recordings of high quality. There are 17 violinists' performances in the collection, captured in a variety of settings. We execute dynamic temporal warping on the constant-Q spectrogram of the synthetic audio and the recording in order to gain the alignment between the two. We synthesise the scores using the open-source synthesizer FluidSynth [23] with MuseScore General SoundFont [24]. In Figure 3, we showcase a sample dataset along with its approximate alignment. We make available to the public the dataset and the code used for the alignment procedure in order to pave the way for future studies on score-to-audio music synthesis. 2 To train the inversion model, we additionally gather an internal dataset of 156 hours of commercial recordings. This is because, as mentioned in Section 2.3, the inversion model does not need aligned data. In addition to the violin dataset, we also take into account the MAESTRO dataset [5], which includes 200 hours of piano recordings from the International Piano-e-Competition [25] over ten years of competition with MIDI recordings that have been finely aligned. Only the synthesis and inversion models can be trained on this dataset, however, since it does not provide the raw scores.

4. EXPERIMENTS & RESULTS

4.1. Implementation details

For the alignment model, we use a three-layer transformer in the encoder. The six transformer layers that make up the decoder in the synthesis model are identical to those in the alignment model. All of our embeddings make use of 128 dimensions. We mimic the network topology of the Hifi-GAN v2 model in [13] for the inversion model. Due of its unique availability, velocity data is only used in the piano dataset. We use the years of the competitions as performer IDs as we do not have performer information for the piano dataset. For the scores, we use a temporal resolution of twenty-four time steps per quarter note. For spectrogram calculation, we use a hop size of 256, which results in a temporal resolution of 16 ms, and downsample the audios to 16 kHz mono. For training, the audios are cut into 5-second segments, with 10% set aside for confirmation. With a batch size of 16, we use the Adam optimizer [26]. We discover that combined training decreases convergence, thus we train the alignment and synthesis models independently, contrary to [21]. All of the synthesis models (100K for violin and 250K for piano) and the alignment model (10K steps) are trained. In every case,



Fig. 4: Both the constant-tempo baseline model and our suggested model, Deep Performer, provide examples of alignments. The input score is shown in (c).

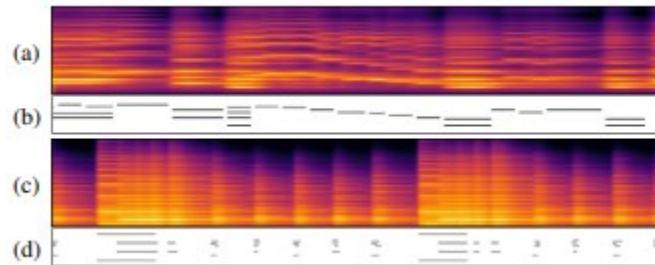


Fig. 5: Our suggested model synthesized examples of the mel spectrograms for (a) the violin and (c) the piano, shown on a log scale. (b) displays the input scores for (a) and (c), whereas (d) shows the same data for (d). various synthesis models share the same dataset and train the inversion model for 1 million steps. We gratefully draw on the code generously made public in [13, 27] for our implementation. In order to process the scores, we use pretty_midi [28] and MusPy [29].

4.2. Qualitative and quantitative results

Figure 4 shows a comparison between the alignments anticipated by our proposed model and those produced by assuming a constant tempo. It is clear that our suggested model can accurately anticipate time and include note rests. Figure 5 shows examples of synthesized mel spectrograms for two polyphonic compositions, demonstrating the efficacy of the proposed polyphonic mixer. In these spectra, distinct harmonic structures and polyphony are visible. Then, we evaluate our suggested synthesis model in comparison to a baseline model that acquires the waveform from an aligned piano roll using a Hifi-GAN [13]. In order to provide an equitable comparison, we furnish a position roll that encodes the note-wise position information and condition this model with the performer IDs. (A position roll is like a piano roll, except that the values go down from 1 to 0 as the note progresses from beginning to conclusion.) Figure 6(a) and (b) show that compared to the baseline model, which provides sharper but noisier results, our suggested model yields more refined harmonic structures and flatter contours, particularly at higher frequencies. The mean squared error (MSE) between the ground truths and the synthesized mel spectrograms is shown in Table 2. Compared to the baseline model, our suggested model obtains a reduced MSE on both datasets. Lastly, our suggested model outperforms the baseline model during training because mel spectrograms have lower temporal resolution than waveforms. On our project website, you may get audio samples.

Table 1: Results of the subjective listening test. The mean opinion scores (MOS) and 95% confidence intervals are reported.

	Violin				Piano
	Pitch accuracy	Timbre	Noise level	Overall	Overall
Hifi-GAN baseline	4.02 ± 0.31	3.13 ± 0.26	2.51 ± 0.29	2.57 ± 0.22	1.49 ± 0.17
Deep Performer (ours)	4.22 ± 0.30	3.26 ± 0.30	2.67 ± 0.31	2.58 ± 0.21	2.17 ± 0.24
- without note-wise positional encoding	4.13 ± 0.29	3.24 ± 0.27	2.52 ± 0.29	2.61 ± 0.23	2.37 ± 0.23
- without performer embedding	3.05 ± 0.52	2.54 ± 0.42	2.04 ± 0.31	2.01 ± 0.25	2.26 ± 0.25
- without encoder (using piano roll input)	4.30 ± 0.36	2.91 ± 0.28	2.39 ± 0.28	2.22 ± 0.18	1.43 ± 0.16

Table 2: Comparisons of the final MSE between the synthesized mel spectrograms and the ground truths, in log scales.

	Violin	Piano
Hifi-GAN baseline	0.892	0.722
Deep Performer (ours)	0.700	0.436
- without note-wise positional encoding	0.700	0.433
- without performer embedding	1.030	0.523
- without encoder (using piano roll input)	0.844	0.621

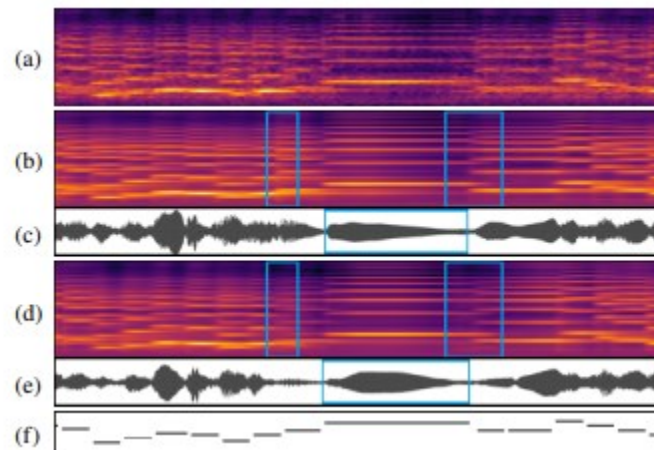


Fig. 6: Examples of the mel spectrograms, in log scale, synthesized by (a) the baseline model, (b) our proposed synthesis model, and (d) our proposed synthesis model without the notewise positional encoding. The corresponding waveforms for (b) and (d) are shown in (c) and (e), respectively. The input score is shown by (f).

4.3. Subjective listening test

We use a subjective hearing test with fifteen people we enlisted from our social networks; fourteen of them are musicians, which allows us to assess our suggested system even further. For each dataset, we choose five musical scores at random and use various models to create a synthesis. On a 5-point Likert scale, participants are asked to assess the synthetic audios based on pitch accuracy, timbre, noise level, and overall quality. Table 1 displays the findings. On the piano dataset, our suggested model accomplishes results that are on par with the baseline, and on the violin dataset, they are much better.

4.4. Ablation study

We examine three stripped-down variants of our model to quantify the roles played by various parts of the suggested model. The first eliminates the note-wise positional encoding. The second one takes the embedded performer out. Keeping the performer embedding, the third variant eliminates the encoder and feeds the decoder piano rolls and position rolls (see to Section 4.2 for more information on this). The highlighted areas in Figure 6(b)–(e) show how note-wise positional encoding improves the model's ability to generate realistic waveform envelopes and transparent note transitions. We also present in Tables 1 and 2 the findings for these ablated models. It is clear that the performance integration greatly enhances the quality on all fronts. Despite the note-wise positional encoding's efficacy, as we demonstrated above, it was not statistically significant in our subjective listening test; this might be because it was swamped by model artifacts. Lastly, using an encoder network yields far better results, indicating that the encoder is capable of learning a more accurate score representation than the piano roll representation.

5. CONCLUSION

To synthesize natural music performance from discordant musical scores, we introduced a new three-stage approach. To match the encoder and decoder with polyphonic inputs, we suggested the polyphonic mixer. To further provide the synthesis model with fine-grained conditioning, we additionally suggested note-wise positional encoding. Our suggested model achieves comparable quality versus the baseline on the violin dataset and substantially outperforms the baseline on the piano dataset, as shown by the subjective listening test. In our future work, we want to include adversarial losses [32, 33] to sharpen the findings, use the articulation marks and decorations on scores to better represent playing approaches [30, 31], and separate timbre from room acoustics to boost controllability [6].

7. REFERENCES

- [1] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, “Neural audio synthesis of musical notes with WaveNet autoencoders,” in ICML, 2017.
- [2] A. Défossez, N. Zeghidour, N. Usunier, L. Bottou, and F. Bach, “SING: Symbol-to-instrument neural generator,” in NeurIPS, 2018.
- [3] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, “GANSynth: Adversarial neural audio synthesis,” in ICLR, 2019.
- [4] R. Manzelli, V. Thakkar, A. Siahkamari, and B. Kulis, “Conditioning deep generative raw audio models for structured automatic music,” in ISMIR, 2018.
- [5] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAESTRO dataset,” in ICLR, 2019.
- [6] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable digital signal processing,” in ICLR, 2020.
- [7] B. Hayes, C. Saitis, and G. Fazekas, “Neural waveshaping synthesis,” in ISMIR, 2021.
- [8] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, “DeepSinger: Singing voice synthesis with data mined from the web,” in KDD, 2020.
- [9] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, “This time with feeling: Learning expressive musical performance,” *Neural Computing and Applications*, vol. 32, pp. 955–967, 2020.
- [10] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions,” in ICASSP, 2018.
- [11] R. Prenger, R. Valle, and B. Catanzaro, “WaveGlow: A flowbased generative network for speech synthesis,” in ICASSP, 2019.
- [12] K. Kumar, R. Kumar, T. de Boissiere, L. Gestein, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, “MelGAN: Generative adversarial networks for conditional waveform synthesis,” in NeurIPS, 2019.
- [13] J. B. Jungil Kong, Jaehyeon Kim, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in NeurIPS, 2010.
- [14] B. Wang and Y.-H. Yang, “PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network,” in AAAI, 2019.
- [15] F. Schimbschi, C. Walder, S. M. Erfani, and J. Bailey, “SynthNet: Learning to synthesize music end-to-end,” in IJCAI, 2019.

- [16] J. W. Kim, R. Bittner, A. Kumar, and J. P. Bello, "Neural music synthesis for flexible timbre control," in ICASSP, 2019.
- [17] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, and J. Engel, "MIDI-DDSP: Detailed control of musical performance via hierarchical modeling," in ICLR, 2022.
- [18] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," arXiv preprint arXiv:2106.15561, 2021.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in NeurIPS, 2017.
- [20] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyriannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in INTERSPEECH, 2017.
- [21] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," in NeurIPS, 2019.
- [22] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei, D. Su, and D. Yu, "DurlAN: Duration informed attention network for multimodal synthesis," in INTERSPEECH, 2020.
- [23] "FluidSynth," <https://www.fluidsynth.org/>.
- [24] "Musescore general soundfont," <https://musescore.org/en/handbook/3/soundfonts-and-sfz-files>.
- [25] "International Piano-e-competition," <https://www.piano-e-competition.com/>.
- [26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in ICLR, 2015.
- [27] C.-M. Chien, J.-H. Lin, C.-Y. Huang, P.-C. Hsu, and H.-Y. Lee, "Investigating on incorporating pretrained and learnable speaker representations for multi-speaker multi-style text-to-speech," in ICASSP, 2021.
- [28] C. Raffel and D. P. W. Ellis, "Intuitive analysis, creation and manipulation of MIDI data with pretty_midi," in ISMIR LateBreaking Demos, 2014.
- [29] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "MusPy: A toolkit for symbolic music generation," in ISMIR, 2020.
- [30] C.-H. Yang, P.-C. Li, A. W. Y. Su, L. Su, and Y.-H. Yang, "Automatic violin synthesis using expressive musical term features," in DAFx, 2016.
- [31] C.-C. Shih, P.-C. Li, Y.-J. Lin, Y.-L. Wang, A. W. Y. Su, L. Su, and Y.-H. Yang, "Analysis and synthesis of the violin playing style of Heifetz and Oistrakh," in DAFx, 2017.
- [32] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in CVPR, 2017.
- [33] J. Yang, J.-S. Bae, T. Bak, Y. Kim, and H.-Y. Cho, "GANSpeech: Adversarial training for high-fidelity multi-speaker speech synthesis," in INTERSPEECH, 2021.